3.0 Summary of Assumptions, Limitations and Errors

This section of ASP I summarizes assumptions, limitations and known errors of ALARM 3.0 derived from any and all applicable sources (especially V&V), and addresses the implications of these for model use or application. This information is useful in helping a user to determine if the model adequately addresses all the radar phenomena and environmental conditions that are important to the intended application. This summary also provides a method for integrating assumptions, limitations, and errors discovered during other V&V efforts.

Details of the assessment procedures for assumptions, limitations, and errors can be found in the *SMART Process Document* [1].

During the course of detailed V&V since 1992, 56 Model Deficiency Reports (MDRs) have been opened against three different ALARM software and documentation baselines. These MDRs address three categories of problems: errors, enhancements needed to support the SMART V&V effort, and general enhancements. These three categories are briefly discussed below.

1. There are 42 MDRs categorized as errors. 13 of those were opened against earlier versions of ALARM, and are now closed with corrections implemented. Of the remaining MDRs, two are withdrawn, three request documentation changes, seven are potentially serious and require further investigation, and the remaining 20 either have fixes identified or are deemed trivial to fix.

   Four of the more serious MDRs involve SEKE, and are awaiting Lincoln Laboratory's (LL's) analysis and recommendations, expected during 1995. Since LL are the inventors and owners of the SEKE algorithms, and understand their code better than anyone else, it is prudent to delay further action on these MDRs until they have completed their review.

   MDR 29, *Incorrect Clutter Processing: Negative Patch Length*, if uncorrected, represents a potentially serious problem to some users. SAIC, however, has written and tested a new clutter subroutine for users needing correct clutter processing.

   MDR 40, *Incorrect Stand-Off Jammer Altitude Reference Type Processing*, identifies a problem for scenarios containing multiple SOJs, with altitude for some being specified as above ground level (AGL) and others as above mean sea level (MSL). This could result in incorrect jammer altitude being used to calculate jamming interference signals.

MDR 44, *Inaccurate Pulse Integration for Scanning Radars*, cites deficiencies in the ALARM implementation of Blake's algorithm for detectability [14]. This could lead to incorrect radar detection performance by the model.

MDR 56, *Noise Jamming Propagation Calculation Error*, reports that ALARM erroneously calculates noise jammer signal strength, which could cause invalid detect/no detect decisions by the model.

2. There are eight changes proposed to model functions which would enhance the utility of the model by allowing direct acceptance of test range data. This category has been primarily driven by V&V assessments using actual test data. Code already exists in several other radar models which could be directly copied or adapted for some requirements of five of these MDRs. Of course, changes which are critical to the evaluation of ALARM have been/will be coded and tested as part of the V&V. Under terms of the Beta Site Agreement, all such changes will be made available to the ALARM CCB.

3. There are six changes proposed to model functions which would enhance the general utility of the model if implemented in the baseline. Of these, code exists for one of them, code is trivial for another two, and major design work is needed for the remaining MDRs.

Assumptions, limitations, and errors are discussed in the following sections. Each is immediately followed by a statement of the implications for model use. Complete MDRs are included in Appendix C.

3.1 Assumptions

In the subsections that follow, assumptions are divided into three groups: data format assumptions, that control the form in which data are input and processed internally; general assumptions, which define how the model emulates overall sensor functionality; and radar function-specific assumptions, which relate to subordinate sensor functionalities within the model. Other, more detailed assumptions may be found in the appropriate subsections of ASP II.

3.1.1 Data Format Assumptions

The following data format assumptions affect the way radar is modeled in ALARM. Except as noted in a few sections below, the source for each assumption is the *ALARM 3.0 User's Manual* [4].

1. Data are input in fixed format within a record (i.e., fields within a record are defined in specific columns). Required records within a DATABLOCK are in fixed order; even optional records can only appear at fixed locations within the DATABLOCK. No comments are permitted within a DATABLOCK or in individual data records. DATABLOCKs, however, can appear in any order within the input stream.

   Implications for Use: This requires careful construction of the input deck. Fortunately, the ALARM output file echoes input to aid the user in determining if the input deck was correctly constructed.

2. Time is input and processed in units of seconds or microseconds, depending on the time variable definition.

   Implications for Use: The user should ensure that input data are specified, and that output data are interpreted, in the correct units.

3. Distances (including altitudes) are input and processed in meters.

   Implications for Use: Since no conversions are provided between meters and feet, statute miles, nautical miles, and data miles, the user should ensure that input data are specified, and that output data are interpreted, in the correct units.

4. Speeds are both input and processed in meters/second.

   Implications for Use: Since conversions are provided between meters/second, knots, and other units, the user should ensure that input data are specified, and that output data are interpreted, in the correct units.

5. Frequencies and bandwidths are input in hertz (Hz) or megahertz (MHz) and processed in hertz.

   Implications for Use: The user should ensure that input data are specified, and that output data are interpreted, in the correct units.

6. Angles are input in degrees and fractions of degrees and are internally processed in radians.

   Implications for Use: None.

7. Locations are input in degrees, minutes, and seconds of latitude and longitude, and processed in radians (angles) and meters (range). Positive numbers represent north latitude and east longitude; negative numbers represent south latitude and west longitude.

   Implications for Use: Because inputs are specified to the nearest second of latitude/longitude, location precision is limited to approximately 30 meters.

8. Temperatures are processed in degrees Kelvin; there are no instances of user-specified input of temperature.

   Implications for Use: None.

9. Each subroutine contains an IMPLICIT statement to force real number default precision in the code to double precision (REAL*8). Source: ALARM 3.0 source code.

   Implications for Use: None.

10. Four-byte integer storage is used throughout ALARM, except for code accessing Defense Mapping Agency (DMA) Digital Terrain Elevation Data (DTED), which is recorded in a two-byte (INTEGER*2) format. Source: ALARM 3.0 source code.

    Implications for Use: None.

11. ALARM uses FORTRAN COMPLEX variables, which internally generate ordered pairs of real variables, the first representing the real part of the complex number, the second, the imaginary part. Each part has the same degree of approximation and the same range as a REAL*4 datum. Source: ALARM 3.0 source code.

    Implications for Use: None.

12. There will be some small differences in the execution of the ALARM code compiled on different computer types; e.g., a SPARC10 versus a PC or a VAX/VMS. Likewise, there could be small differences on the same machine type (e.g., a PC) using a DOS compiler versus a UNIX compiler. These differences arise from several sources: different floating point devices in the hardware, which process specific mathematical functions using differing algorithms or different precision; differing sets of software functions (in the compiler library) or hardware coprocessor calls; errors in the implementation of both hardware and software; and differences in the compilers' optimizer options.

    Implications for Use: Based on review of ALARM outputs from model runs made on several different computers in ECSRL, any such differences should be minor (not detectable in discrete data before four or five digits to the right of the decimal point).

    There are two specific problems related to this assumption. First, there is no standard random number generator (RNG) in the formal definition of FORTRAN 77. The baseline is designed around the VMS system library RNG called RAN. This means that users on other operating systems (i.e., non-VAX/VMS) must change the baseline to call the RNG on their system, which may produce differences in stochastic runs of the model.

    One alternative for avoiding this problem would be to build a dedicated, ALARM-specific RNG, distributed as part of the baseline code. Another alternative would be for the user to build an RNG for his specific operating system/platform which produces random number streams equivalent to those produced by VAX/VMS RAN.

The second problem is found in the implementation of the FORTRAN implicit functions library in the user's compiler. In some compilers, the library returns the maximum precision available on the numeric data processor, typically double precision. Thus, if the results of a function call should be limited to single precision, the parameter may be returned as double precision.

3.1.2 General Assumptions

General assumptions which affect the way radar is modeled in ALARM include the following:

1. Analysts using ALARM should be thoroughly familiar with ground-based radar systems, so that the user-supplied input blocks are reasonable. Source: *Software User's Manual* [4].

   Implications for Use: Input consists of engineering-level data such as transmit power, pulse width, pulse repetition frequency, antenna gain patterns and RCS tables, and data needed to simulate MTI and pulse doppler processing. ALARM checks that the input data blocks are complete and that values are within specified limits; these data are not evaluated for engineering-level validity, however.

2. ALARM has been designed to perform initial target detection determination by a single radar. In the Flight Path mode, ALARM target location inputs merely specify the locations at which initial detection will be determined. ALARM does not model any tracking algorithm. It does not consider the capabilities of a human operator. It does not model an automatic plot extractor. It does not adjust the probability of detection (P_d) for detection maintenance after the initial detect. Source: *Software Analyst's Manual* [6].

   Implications for Use: The modeler must carefully compare the study requirements with capabilities offered in the model.

3. Calculations are performed using power ratios, not voltage levels. Source: *Software Analyst's Manual* [6].

Implications for Use: Validation efforts using data recorded in units of voltage must be converted for correct usage.

4. The analyst is restricted to one-vs-one scenarios. Source: *Software Analyst's Manual* [6].

   Implications for Use: ALARM may not be the appropriate tool for complex multi-aircraft scenarios.

5. In the Contour Plot mode, ALARM inputs do not allow the user to specify pitch and roll angles. Source: *Software User's Manual* [4].

   Implications for Use: Analysts studying the detection of aircraft which normally fly in an attitude different from that in which the RCS was taken cannot accurately portray their targets.

6. ALARM incorporates the Swerling/Barton technique to account for detectability of different classes of fluctuating targets. Source: *Software Analyst's Manual* [6].

   Implications for Use: If the analyst cannot identify the correct distribution to represent a fluctuating target aircraft, the model results may be incorrect.

7. ALARM provides a limited capability to simulate continuous wave (CW) radar systems. CW is modeled as a special case of the pulse doppler radar, and uses the following parameters described below. Source: *Software Analyst's Manual* [6].

   a. for pulse blanking/eclipsing, a bistatic radar is assumed and the eclipsing loss is set to 0.0 dB;

   b. a duty cycle of 1.0 is used;

   c. only one PRF is allowed;

   d. average power is used;

   e. the pulse width is set to 1/PRF.

Implications for Use: The user should carefully consider whether ALARM is suitable for modeling the CW radar system needed in the user’s application.

8. While the primary purpose of the model is the detection of low-altitude targets, higher altitude targets can also be used, with the limiting factor that only one radar beam can be modeled. Source: *Software Analyst’s Manual* [6].

   Implications for Use: For multiple beam radars, modeling only the lowest beam is in most cases adequate for evaluating the radar's capability against low-flying targets. For higher altitude targets, modeling the low beam should provide the maximum detection capability of the radar; however, gaps may appear close to the radar since this area is normally covered by higher-angle beams. ALARM can model the scan coverage of a tracking antenna beam by specifying a minimum and maximum elevation angle. This in effect simulates a continuously scanning beam. This capability can be used to closely simulate a stepped-scan radar with the assumption that the target is always detected by the peak of the beam. There are no provisions, however, for changing the radar characteristics as the beam scans in elevation. While ALARM can only model one beam of a radar at a time, multi-beam radars can be modeled through multiple runs, one for each beam, and use of the off-line utility program PDMERG.

   Through careful preparation of the run inputs and use of the off-line utility program, ALARM can be used to model the more complicated ground-based radars. The user should carefully consider whether ALARM is suitable for modeling the particular radar system needed in the user’s application.

3.1.3 Function-Specific Assumptions

There are many assumptions specifically related to particular functions within the model:

1. The mean earth radius at sea level is assumed to be a constant, 6,371,007 meters. Source: ALARM 3.0 Source Code, subroutine INITCT.

Implications for Use: Use of a single value implies that the earth is a perfect sphere, not an oblate spheroid. If a study requires a more precise definition of the earth's shape at a given site, the analyst must provide the appropriate radius.

2. All terrain within the volume illuminated by the radar is assumed to have the same land form, land cover, roughness factor, dielectric constant, ground conductivity, and refractivity. Source: *Software Analyst's Manual* [6].

   Implications for Use: For sites where the same values for these factors are not appropriate (e.g., a radar with coverage split between land and ocean), the analyst must either make multiple runs, splitting the flight path; select another model; or modify the baseline to accommodate non-uniform terrain characteristics, including modeling radars sited near the coast.

3. Only single bounce (in each direction) specular reflection paths are used for calculating multipath. Neither multiple bounce paths nor diffuse reflections are considered. Source: *Software Analyst's Manual* [6].

   Implications for Use: For low-altitude targets flying over very rough terrain, the multipath propagation factor (FTO4TH) may be different than measured field test data, even though the SEKE diffraction/multipath propagation algorithm was developed from empirical data and should be correct. LL has expressed concern over the accuracy of the multipath implementation in ALARM and is currently evaluating the code.

4. No radar ducting is modeled in ALARM; i.e., the propagation factor (RKFACT) must be greater than zero. Source: *Software User's Manual* [4].

   Implications for Use: If the study requires over-water detection analysis in a geographic area where the radar signal is normally affected by ducting, the analyst must either select another model or modify the baseline to accommodate unique study needs.

5. The propagation factors generated for the path from the radar to and from any point in space are identical. Source: *Software Analyst's Manual* [6].

Implications for Use: Real-world effects may not differ significantly, depending upon the terrain. For some cases, however, the analyst must either select another model or modify the baseline to accommodate unique study needs.

6. The propagation factor is calculated using only those terrain points that lie in the vertical plane defined by the radar, the target, and the center of the earth. Propagation effects due to points outside this plane are not considered. Source: *Software Analyst's Manual* [6].

   Implications for Use: The resultant signal using the modeled propagation factor may differ significantly from test range data. Depending upon terrain roughness, the analyst must select another model or modify the baseline to accommodate unique study needs.

7. ALARM uses a perfectly pointed radar receive antenna. Source: *Software Analyst's Manual* [6].

   Implications for Use: For tracking radars, this means radar signal strength is calculated without the effect of tracking error. This is not a problem for ALARM users, since ALARM does not model tracking radars. For early warning (EW), acquisition (ACQ), or other fixed-scan radars, this means that the radar is pointed directly at the target in azimuth. Real EW or ACQ radars are constantly scanning in azimuth and generally do not point directly at the target at the time the target is first susceptible to detection.

   If antenna pointing error is known for a given set of collected test data, the model can be easily modified to input that error for each flight path point; otherwise, the analyst must select another model or enhance the baseline to accommodate unique study needs.

8. ALARM models stand-off jammers with a single value representing gain and power from a single fixed location. Source: *Software User's Manual* [4].

   Implications for Use: This assumption probably came from the concept of a stand-off jammer operating in a strategic scenario, at long distances from the radar. In such a geometry, the location of the jammer would appear to not move over several minutes of a scenario. Antenna gain and power would not change significantly, even with a steerable antenna. Modern scenarios tend to be tactical in nature, possibly requiring penetrating jamming platforms (e.g., EA-6B or EF-111A) and agile jamming techniques.

Implications for Use: The analyst is constrained to using static stand-off noise jammers. If the study requires moving stand-off jamming, the analyst must enhance the baseline to accommodate unique study needs or select another model.

9. The ALARM radar receiver is linear in response over the full range of inputs; i.e., a change in input signal level directly affects the receiver output signal. Dynamic range of the receiver is not considered in the model's algorithms. Realistic automatic gain control (AGC) is not modeled. Source: *Software Analyst's Manual* [6].

   Implications for Use: The modeled radar is not sensitive to the effects of finite dynamic range. If, however, AGC is an important consideration of the study, the analyst must select another model or enhance the baseline to accommodate unique study needs.

10. ALARM considers the target to be a single point, located at the center of gravity (COG) of the target. Source: *Software Analyst's Manual* [6].

    Implications for Use: Since the model uses perfect pointing of the receive antenna, there is no detrimental effect on the calculations caused by this target representation. However, real radars point imperfectly at their targets. Actual tracker performance varies with radar and target type, but rarely will the perceived center of the actual target coincide with the target's COG. Perfect antenna pointing, especially for a tracking radar, means that the modeled radar performance in ALARM is better than that normally experienced with an actual radar.

11. ALARM uses atmospheric attenuation data derived only for a Standard Day. Source: *Software Analyst's Manual* [6].

Implications for Use: Temperature and humidity can significantly affect the amount of radar signal attenuation experienced. For desert or arctic conditions, an analyst should consider installing different attenuation values in the tables in subroutine ATTEN.

12. ALARM assumes that MTI voltage amplitudes at the sum circuit are equal. Source: *Software Analyst's Manual* [6].

    Implications for Use: This is generally not true in real radar systems, where the cancellation of stationary signals (e.g., clutter) is less than perfect. ALARM's detection performance in clutter is, therefore, optimistic. Furthermore, in a real radar, the delay-line canceler may be frequency weighted, causing invalid response from the MTI equation used in ALARM to occur. The analyst must either select another model or enhance the baseline to accommodate unique study needs. The Center for Naval Analyses has modified and documented changes to an older version of ALARM to accommodate an adaptive MTI. A similar change could be used to meet MTI requirements of a study.

3.2 Limitations

There are several kinds of limitations in ALARM:

1. computer system-related limits (real number arithmetic, byte/word size, etc.);
2. embedded parameter limits in the code;
3. test data-related functional limits; and
4. functional limits (not all functional elements are fully implemented in any model).

These are discussed in detail in the following sections.

3.2.1 Computer System-Related Limitations

These are generally dealt with by the vendor of the FORTRAN compiler by adherence to some external standards. Trivial examples include the limits of values which can be portrayed in INTEGER or REAL variables.

Implications for Use: None.

3.2.2 Parameter Limitations

PARAMETER statements in the FORTRAN code, such as the maximum number of flight path points, may be changed by the user. Typically this requires the use of the utility program DIMENS to redefine the parameter, followed by recompiling and relinking the ALARM code. table 3.2-1 identifies and describes the parameter statements used in ALARM. Source: Review of ALARM 3.0 source code.

Table 3.2-1 ALARM Parameters

Parameter Name	Default Value	Description
MAXBUF	48	Maximum number of DMA DTED input buffers.
MAXCOL	3	Number of DMA DTED cells per row.
MAXDIM	256	Maximum number of locations in X and Y directions in the Contour Plot mode.
MAXNTP	512	Maximum number of points in target flight path data.
MAXPRF	4	Maximum number of radar pulse repetition frequencies (PRFs).
MAXRES	10	Number of terrain cells separating a secondary knife edge from the main knife edge.
MAXROW	2	Number of DMA DTED cells per column.
MAXTRP	25	Maximum number of iterations in the Romberg trapezoidal approximation.
MAZFLC	36	Maximum number of target RCS fluctuation data azimuth sectors.
MAZRCS	721	Maximum number of target RCS data azimuth sectors.
MELFLC	18	Maximum number of target RCS fluctuation data elevation sectors.
MELRCS	361	Maximum number of target RCS data elevation sectors.
MKNIFE	2	Maximum number of secondary knife edges in terrain profile.
MLOCAL	128	Maximum number of local terrain maxima in terrain profile.
MOPENF	5	Maximum number of open DMA DTED terrain files.
MPROFL	2048	Maximum number of points in the terrain profile.
MXACOL	1	Number of elevation cuts in antenna gain patterns.
MXAZEL	1800	Number of azimuthal cuts in antenna gain patterns.
MXBRCS	10	Maximum number of rotor blade RCS data.

Table 3.2-1 ALARM Parameters

Parameter Name	Default Value	Description
MXFEAT	2	Number of points on either side of a terrain point in the terrain profile for that point to be considered a local maximum.
MXGATE	4	Maximum number of MTI gates.
MXNJAM	6	Maximum number of jammers.
MXNVAR	50	Maximum number of variables recorded in Flight Path mode output array.
NBLOCK	13	Number of input DATABLOCK types. Used for system level control of the model and should not be changed by a user.
NITERA	256	Maximum number of iterations in calculation of P_d, given P_{fa} and target signal.
PDTOLR	10^{-3}	Used with NITERA, an iteration tolerance value.

Implications for Use: Some limitations are non-trivial; three cases are discussed below.

1. The number of elevation cuts in the antenna gain patterns (MXACOL) must be increased to allow ALARM to handle three-dimensional antenna patterns. MXACOL is used as one dimension in the declaration of the antenna gain arrays. To increase this parameter indiscriminately brings the risk of generating antenna gain arrays so large that the program does not easily fit in available memory. If a three-dimensional antenna gain pattern does not significantly affect the outcome of the study, as in a high clutter and/or jamming environment, the two-dimensional patterns should be used.

2. The number of DMA DTED input buffers directly affects the model’s execution time. If the user has large amounts of memory available, then increasing the default number of buffers (48) could improve the execution time. Conversely, decreasing the number of buffers (so that the executable code can fit in limited memory) can backfire. For a user with only 2MB of memory in his computer, SAIC decreased the number of buffers from 48 to one. When that source code was later moved to an Alliant super-minicomputer, an example case, which should have taken just a few minutes to complete, ran for over 24 hours. When the user’s computer system analyst investigated the problem, he discovered that the job was almost completely I/O-bound; i.e., the program spent most of the time processing input/output requests. Upon restoring the baseline number of buffers, the job ran normally.

3. The number of ALARM input DATABLOCK types (NBLOCK) is used as an executive-level datum by the model. This value should normally only be changed with the addition of some functionality requiring a new input data block type.

3.2.3 Test Data-Related Functional Limitations

Some functional limits prevent the direct use of range test data. These have generally meant that efforts to perform sensitivity analyses or to validate functional elements have required off-line programs to translate data into a useful format, modification of ALARM subroutines to read the test range data directly, addition of some new functionality to determine if that function is needed in the model, or some combination of the above. Range test data limitations include ALARM's inability to input Cartesian coordinates (as are found in most range Time Space Position Indicator (TSPI) data).

Such range test data-related limitations have resulted in the model deficiency reports (MDRs) identified in table 3.2-2. Model enhancements to address the MDRs in this table would also present more utility to the general ALARM user community as well.

Table 3.2-2 Model Deficiency Reports - Requested Improvements for the Use of Range Data

MDR	Date	Description	Disposition
14	4 Dec 92	Greater than ±30 m Flight Path mode location accuracy	Users
15	15 Dec 92	Implement Flight Path mode Cartesian coordinates	Users
16	15 Dec 92	Add flight paths for stand-off jammers	Users
17	15 Dec 92	Add transmitter antenna patterns for stand-off jammers	Users
23	11 Jan 94	Add internal binary data file documentation	New
25	26 Jan 94	Add STC processing	New
26	26 Jan 94	Add measured clutter processing	New
39	4 Oct 94	Add CFAR processing	New

Disposition:
Users - Referred to Users group for possible future implementation.
New - Not yet evaluated for implementation.

3.2.4 Other Functional Limitations

More general functional limitations can be found in functions implemented in some restricted manner in the model, or result from user requests for functions not yet implemented in ALARM. Sensitivity analyses and functional element validation have illuminated several such areas. These have been documented in MDRs identified in table 3.2-3 below.

Table 3.2-3 Model Deficiency Reports - General Requested Improvements

MDR	Date	Description	Disposition
10	9 Nov 92	Add MSL altitude option for Contour Plot mode	ALARM 92
12	9 Nov 92	Add ambient air temperature processing	N/A
13	9 Nov 92	Add plot size control option	Users
28	11 Feb 94	Add atmospheric attenuation for more categories of atmospheric conditions	New
32	2 Jun 94	Add an ALARM-internal random number generator	New
38	15 Sep 94	Add target pitch vector for Contour Plot mode	New

Disposition:
ALARM 92 - Implemented in ALARM92 (Beta version).
Users - Referred to Users group for possible future implementation.
N/A - Not implemented/not required.
New - Not yet evaluated for implementation.

3.3 Errors and Anomalies

Recent sensitivity analyses and V&V efforts have revealed a number of errors in the ALARM source code and documentation. ALARM MDRs have been opened to document these, while some additional errors found during verification are still under review. The MDRs have been submitted for action to ECSRL at Wright-Patterson AFB, OH, the organization responsible for the ALARM code.

MDR 8, *Is Atmospheric Attenuation already included in SEKE?* was withdrawn after investigation revealed it was not.

MDR 2, *Incorrect Pulse Blanking*, represents a point of unresolved technical disagreement within the ALARM user community.

The remaining MDRs have either been corrected in ALARM 92 or ALARM 3.0, or they have not yet been considered for inclusion in a formal baseline update. Under current plans, the next ALARM baseline is tentatively funded/scheduled for late FY95.

Major MDR error reports are identified in table 3.3-1 and are also included at Appendix C.

Table 3.3-1 Model Deficiency Reports - Errors

MDR	Date	Description	Disposition
1	10 Jul 92	Incorrect staggered pair MTI calculations	ALARM 3.0
2	10 Jul 92	Incorrect pulse blanking	TBD
3	10 Jul 92	Square antenna aperture gain approximation	ALARM92

Table 3.3-1 Model Deficiency Reports - Errors

MDR	Date	Description	Disposition
4	10 Jul 92	Incorrect pulse compression	ALARM 3.0
5	12 Aug 92	Atmospheric attenuation execution error- array bounds	ALARM 3.0
6	12 Aug 92	Miscellaneous documentation errors	ALARM 3.0
7	12 Aug 92	Inconsistent input error checking: land form/cover	ALARM92
8	9 Nov 92	Is atmospheric attenuation already included in SEKE?	Withdrawn
9	9 Nov 92	Negative round earth diffraction factors	ALARM92
11	9 Nov 92	Atmospheric attenuation formula error	ALARM92
18	12 Mar 93	Plot file overwrite flag error	ALARM92
19	15 Mar 93	Incorrect antenna gain interpolation	ALARM92
20	11 Aug 93	Atmospheric attenuation error at 16 GHz	ALARM 3.0
21	21 Oct 93	Incorrect SEKE antenna gain calculation	New
22	11 Jan 94	Incorrect clutter processing for coastal sites	New
24	26 Jan 94	Incorrect SEKE processing (per Lincoln Laboratory)	New
27	31 Jan 94	Add SEKE1 algorithm	New
29	6 Apr 94	Incorrect clutter processing: negative patch length	New
30	26 Apr 94	Terrain database set up error	New
31	2 Jun 94	Insufficient limits on P_{fa} and P_d	New
33	2 Jun 94	Incorrect sea-state definitions in the *Analyst's Manual*	New
34A	21 Sep 94	Incorrect SEKE diffraction affects threshold	New
35	26 Aug 94	Superfluous code in subroutine CLUTPD	New
36	26 Aug 94	Inconsistent treatment of doppler processing	New
37	12 Sep 94	Incorrect flight path data echo print	New
40	29 Dec 94	Incorrect stand-off jammer altitude reference type processing	New
41	10 Feb 95	Incorrect ratio for radar site	New
42	10 Feb 95	Incomplete error checking of BLUEMAX format DATATARG inputs	New
43	24 Feb 95	Incorrect pulse integration for visual detection	New
44	24 Feb 95	Inaccurate pulse integration for scanning radars	New
45	9 Mar 95	Possible overflow errors in subroutine THRESH	New
46	9 Mar 95	Incorrect print format in subroutine RCSPRT	New
52	9 Mar 95	Incorrect end-of-file processing in subroutine RDRINP	New
56	9 Mar 95	Noise jamming propagation calculation error	New

Disposition:
ALARM 92 - Implemented in ALARM92 (ALARM 3.0 beta version).
ALARM 3.0 - Implemented in ALARM 3.0.
New- Not yet reviewed by the ALARM Users Group and CCB
TBD - Disposition unresolved.

Implications of these errors for model users are discussed below.

1. All MDRs identified by a status indicating an implementation version have been fixed in the specified version.

2. MDRs 21, 24, 27, and 34A represent identified differences between the ALARM implementation of the Lincoln Laboratory (LL) SEKE propagation algorithms, and the LL version. On several occasions LL has briefed differences in the propagation factor generated in ALARM/SEKE vice that generated by their in-house SEKE code. To date, LL has specifically observed that the majority of these differences seem to be caused by the multipath calculations. Both LL and SAIC are committed to reviewing the SEKE code during FY95. Corrections to the ALARM implementation are anticipated pending the results of those investigations.

3. MDR 22, *Incorrect Clutter Processing for Coastal Sites*, will only cause problems in the modeling of radar performance of coastal sites where the target appears both over land and over water. This problem can be dealt with by making two separate model runs with different propagation/clutter parameters, then manually merging the results. Code has been developed by SAIC to provide the data to the model to more accurately represent the problem, but it is not yet known whether the CCB will approve the proposed change.

4. MDR 29, *Incorrect Clutter Processing: Negative Patch Length*, can be a significant problem in modeling high-frequency radars in clutter environments. Depending on the radar’s range to the target, the clutter signal will fluctuate erratically over short distances in a flight path, between some nominal value (e.g., -100 dB) and the ALARM minimum signal level (-380 dB). If the study involves some sensitivity to the clutter signal level, this MDR identifies a serious liability to that study. SAIC has available several alternative clutter subroutines which have been shown to alleviate the error. It is not yet known whether the CCB will approve the proposed change.

5. MDR 30, *Terrain Database Set Up Error*, causes the model to run in round smooth earth mode if the terrain database echo print request flag (IPRTDB) is not set in the input stream. This problem can be easily avoided by ensuring that the print request flag is always turned ON (=1). Otherwise, no terrain data is actually used for the run.

6. MDR 31, *Insufficient Limits on P_{fa} and P_d*, identifies the need for additional validity checks on user input values for P_{fa} and P_d. The error will not affect most ALARM runs. The algorithm is invalid only if the user specifies a probability of detection outside the range [0.5,0.99], or a probability of false alarm outside the range [10^{-12},10^{-4}]. The proposed correction to the problem would install legality checks to reject user input outside those ranges.

7. MDR 33, *Change Sea State Definition in Documentation,* suggests clarification of the actual definitions of the sea states used in the model. There is no impact on the informed user.

8. MDR 35, *Superfluous Code in CLUTPD*, identifies a problem in the code wherein the second half of an IF-THEN-ELSE construct could never be executed. Analysis of the code shows that the design of ALARM calls for using a Romberg approximation to integrate the clutter spectral density. The extant ALARM 3.0 baseline, however, neglects the discontinuities in the function at $\pm 3\sigma$. The magnitude of this error should be minimal, less than one percent of the area under the curve. While this MDR represents a legitimate problem, the significance is minimal, and can be ignored by the user.

9. MDR 36, *Inconsistent Treatment of Doppler Processing*, may have a minor effect on a target with a very slow angular target speed and doppler signal components close to zero Hz from all sources. In such cases, the signal may not be automatically dropped by the first doppler filter (the one around 0 Hz). Impact on the user is not known.

10. MDR 37, *Incorrect Flight Path Data File Echo Print*, identifies some minor output format errors in the subroutine TGTPRT. If the user is operating ALARM in the Flight Path mode, care should be taken to remember that the ‘N’-orth latitude designator is not printed for target locations north of the equator, and the ‘S’-outh latitude designator may be erroneously printed for targets crossing from the southern hemisphere to the northern hemisphere. Otherwise, this problem is inconsequential.

11. MDR 40, *Incorrect Stand-Off Jammer Altitude Reference Type*, reports that ALARM 3.0 incorrectly reads an SOJ altitude reference type (either AGL or MSL) into a scalar variable rather than an array indexed by the jammer number. The effect is to use the altitude reference type of the last jammer data input as the type for all SOJ jammers. Incorrect jammer altitude could be used in multiple stand-off jammer scenarios to calculate jamming interference

signals. If line-of-sight due to terrain masking were denied or allowed inappropriately, jamming signal levels could be significantly affected.

12. MDR 41, *Incorrect Ratio for Radar Site*, states that ALARM 3.0 incorrectly sets the value of the variable DRATIO(0) to the user-specified height of the antenna; it should be set to zero. The impact of this error is currently unknown.

13. MDR 42, *Incomplete Error Checking of BLUEMAX Inputs*, notes that ALARM 3.0 fails to completely check the BLUEMAX-format time field. Since this value is never used in ALARM, there is no impact on the user.

14. MDR 43, *Incorrect Pulse Integration for Visual Detection*, points out that ALARM supports only automatic (non-visual) detection. There is no impact on the knowledgeable user.

15. MDR 44, *Inaccurate Pulse Integration for Scanning Radars*, describes deficiencies in the calculation of the S/I threshold related to the number of pulses integrated. This is a potentially serious problem; unknowing, incorrect user specification of the effective number of pulses integrated could lead to incorrect radar detection performance.

16. MDR 45, *Possible Overflow Errors in Subroutine THRESH*, points out possible problems caused by user input of very small values for chi-squared distribution variables. Depending on the operating system/hardware, this could result in a program abort or continued operation without notification to the user of the error. To avoid overflows: (1) for a chi-squared target fluctuation type, the user must specify a non-zero value for the CORELB and CHINDF variables; (2) for a Weinstock fluctuation type, the user must specify a non-zero value for CHINDF.

17. MDR 46, *Incorrect Print Format in Subroutine RCSPRT*, documents problems with printing of output data for scenarios having more than four target fluctuation elevation sectors. The impact of this MDR is minor, as RCSPRT simply echoes input RCS values.

18. MDR 52, *Incorrect End-of-File Processing in Subroutine RDRINP*, reports final checking of the DATARADR input block is bypassed. Errors involving incomplete DATARADR input blocks are not detected by ALARM 3.0.

19. MDR 56, *Noise Jamming Propagation Calculation Error*, reports that ALARM erroneously calculates noise jammer signal strength, which could cause invalid detect/no detect decisions by the model.